

Laborator 3

1. Implementati mergesort si quicksort (gasiti prezentarea algoritmilor si pseudocodul in cursul 2).
2. Analizati functiile implementate mai sus. Care reprezinta metode de sortare stabile? Care e complexitatea? Alegeti un vector cu 6 elemente si realizati pe foaie pasii din cei doi algoritmi de sortare.
3. **(Bonus)** Exista algoritmi de sortare care nu se bazeaza pe comparatii (ca cei de la pctul 1). Acestia se numesc algoritmi de tip „counting sort” si au complexitate temporara redusa $< O(n \log n)$. Implementati. Care e complexitatea spatiala si cea temporală? De ce nu este mai folosit?

Counting sort pe scurt

- algoritm ce se bazeaza pe faptul ca valorile de sortat se gasesc intr-un anumit domeniu.
- functioneaza bazandu-se pe ideea de a numara de cate ori apare fiecare valoare din domeniul admisibil de valori in vector
- pe baza frecventei de aparitie se calculeaza pozitiile elementelor in cadrul vectorului de iesire (sortat)

Exemplu: Fie domeniul admisibil 0...9. Vectorul de sortat e v ($n=7$): 1 9 4 1 1 9 6.

Se foloseste un vector de frecvente - vf (contine numarul de aparitii al fiecarei valori din vectorul de sortat), initial vf are toate elementele 0. Dimensiunea lui vf e egala cu cea a domeniului admisibil de valori:

vf - index: 0 1 2 3 4 5 6 7 8 9

- continut: 0 0 0 0 0 0 0 0 0 0

Se analizeaza numarul de aparitii pentru fiecare valoare din v . vf devine:

vf - index: 0 1 2 3 4 5 6 7 8 9

- continut: 0 3 0 0 1 0 1 0 0 2

Se modifica v astfel incat sa stocheze valorile in ordine: 1 1 1 4 6 9 9.